

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In Re U.S. Patent Application )

Applicant: Wataru Ishisaki )

Serial No. )

Filed: August 26, 1999 )

For: MENU SYSTEM REQUIRING )  
REDUCED USER )  
MANIPULATION OF AN )  
INPUT DEVICE )

Art Unit: )

I hereby certify that this paper is being deposited with the  
United States Postal Service as Express Mail in an  
envelope addressed to: Asst. Comm. for Patents,  
Washington, D.C. 20231, on this date

08/26/99  
Date

Express Mail Label No.:  
EL409487555US

CLAIM FOR PRIORITY

Assistant Commissioner for Patents  
Washington, DC 20231

Sir:

Applicant claims foreign priority benefits under 35 U.S.C. § 119 on the  
basis of the foreign application identified below:

Japanese Patent Application No. 11-034154

A certified copy of the priority document is enclosed.

Respectfully submitted,

GREER, BURNS & CRAIN, LTD.

By

Jonathan D. Feuchtwang  
Reg. No. 41,017

August 26, 1999  
Sears Tower - Suite 8660  
233 South Wacker Drive  
Chicago, IL 60606  
(312) 993-0080

Atty. General 1999-03-00-0  
#3

日 本 国 特 許 庁  
PATENT OFFICE  
JAPANESE GOVERNMENT

JP 09/384073  
08/26/99

別紙添付の書類に記載されている事項は下記の出願書類に記載されて  
る事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed  
this Office.

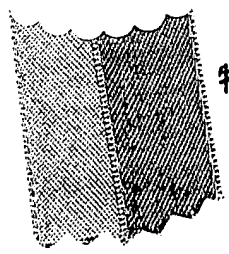
願 年 月 日  
Date of Application: 1999年 2月12日

願 番 号  
Application Number: 平成11年特許願第034154号

願 人  
Applicant(s): 富士通株式会社

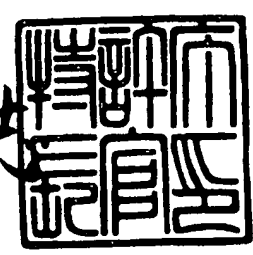
CERTIFIED COPY OF  
PRIORITY DOCUMENT

1999年 6月18日



特許庁長官  
Commissioner,  
Patent Office

山 建 志



【書類名】 特許願

【整理番号】 9950051

【提出日】 平成11年 2月12日

【あて先】 特許庁長官殿

【国際特許分類】 G06F 3/14  
G06F 15/00

【発明の名称】 コンピュータシステム

【請求項の数】 10

【発明者】

    【住所又は居所】 富山県婦負郡八尾町保内二丁目 2 番 1 株式会社富山富士通内

    【氏名】 石崎 互

【特許出願人】

    【識別番号】 000005223

    【氏名又は名称】 富士通株式会社

【代理人】

    【識別番号】 100072590

    【弁理士】

    【氏名又は名称】 井桁 貞一

    【電話番号】 044-754-3035

【手数料の表示】

    【予納台帳番号】 011280

    【納付金額】 21,000円

【提出物件の目録】

    【物件名】 明細書 1

    【物件名】 図面 1

    【物件名】 要約書 1

    【包括委任状番号】 9704486

【ブルーフの要否】 要

【書類名】 明細書

【発明の名称】 コンピュータシステム

【特許請求の範囲】

【請求項 1】

入力デバイスの操作に基づき表示画面上にメニューを表示するコンピュータシステムにおいて、

メニューに表示する項目をグルーピングして管理する管理手段と、

入力デバイスの特定の操作回数または特定の操作の継続時間に基づき、前記グルーピングしたメニュー項目を切り替えて表示させる表示制御手段とを有することを特徴とするコンピュータシステム。

【請求項 2】

入力デバイスの操作に基づき表示画面上にメニューを表示するコンピュータシステムにおいて、

メニューに表示する項目をグルーピングして管理する管理手段と、

オペレーティングシステムから入力デバイスの操作による特定のイベントが発生した旨のメッセージを受信する受信手段と、

当該イベントの発生回数に基づき、前記グルーピングしたメニュー項目を切り替えて表示させる表示制御手段とを有することを特徴とするコンピュータシステム。

【請求項 3】

請求項 2 のグルーピングは、表示するメニュー項目の機能毎にグルーピングされていることを特徴とする請求項 2 のコンピュータシステム。

【請求項 4】

請求項 2 の表示制御手段は、一定時間以内に、入力デバイスの操作が同一点で操作された場合に起動されることを特徴とする請求項 2 のコンピュータシステム。

【請求項 5】

請求項 2 の表示制御手段は、一定時間以内に、入力デバイスの操作が一定の範囲内で操作された場合に起動されることを特徴とする請求項 2 のコンピュータシステム。

【請求項 6】

入力デバイスの操作に基づき表示画面上にメニューを表示するコンピュータシステムにおいて、

メニューに表示する項目をグルーピングして管理する管理手段と、

オペレーティングシステムから入力デバイスの操作による特定のイベントが発生した旨のメッセージを受信する受信手段と、

当該イベントの継続時間に基づき、前記グルーピングしたメニュー項目を切り替えて表示させる表示制御手段と

を有することを特徴とするコンピュータシステム。

【請求項 7】

請求項 6 のグルーピングは、表示するメニュー項目の機能毎にグルーピングされていることを特徴とする請求項 6 のコンピュータシステム。

【請求項 8】

請求項 6 の表示制御手段は、一定時間以内に、入力デバイスの操作が同一点で継続して操作された場合に起動されることを特徴とする請求項 6 のコンピュータシステム。

【請求項 9】

入力デバイスの操作に基づき、メニューをコンピュータの表示画面上に表示させるコンピュータプログラムを記憶したコンピュータ読取り可能な記憶媒体において、

コンピュータのオペレーティングシステムから入力デバイスの操作による特定のイベントが発生した旨のメッセージを受信するプログラムコード手段と、

当該イベントの発生回数に基づき、コンピュータシステム上に生成されるグルーピングされたメニュー項目を切り替えて表示させるプログラムコード手段と

を有することを特徴とするコンピュータ読取り可能な記憶媒体。

【請求項 10】

入力デバイスの操作に基づき、メニューをコンピュータの表示画面上に表示させるコンピュータプログラムを記憶したコンピュータ読取り可能な記憶媒体において、

コンピュータのオペレーティングシステムから入力デバイスの操作による特定のイベントが発生した旨のメッセージを受信するプログラムコード手段と、

当該イベントの継続時間に基づき、コンピュータシステム上に生成されるグルーピングされたメニュー項目を切り替えて表示させるプログラムコード手段とを有することを特徴とするコンピュータ読取り可能な記憶媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、コンピュータシステムの改良されたグラフィカル・ユーザ・インターフェース（GUI）に関し、特に、入力デバイスの操作内容に基づき、コンピュータシステムの画面上にメニューを表示させる場合の、改良されたグラフィカル・ユーザ・インターフェースを提供するものである。なお、以下において、コンピュータシステムの画面を、単に、デスクトップという場合がある。

【0002】

【従来の技術】

第3図から第5図は、入力デバイスの操作内容に基づき、デスクトップ上にメニューを表示する場合の、従来のGUIを表す。

【0003】

第3図は、第1の先行技術のGUIを示す。このGUIは、Microsoft社のWindowsシステム（WindowsはMicrosoft社の登録商標である）やX OpenリミテッドのX Windowシステム（X Windowは、X Openリミテッドの登録商標である）等のオペレーティングシステム、種々のウィンドウ化されたアプリケーション・プログラムで提供されている。同図において、300はデスクトップを、310はアプリケーション・プログラム、文書等のオブジェクトを表すアイコンを、320はポインティングデバイスのマウス・カーソルを、330はポップアップ・メニューを表す。同図は、ユーザが、デスクトップ上若しくはアプリケーションプログラムのウィンドウ内の何も無い領域上または特定の操作可能なオブジェクト上（以下、「何も無い領域／オブジェクト上」という）で、マウスの右クリックをすると、そのマウス・カーソル320の位置にポップアッ

メニュー 330 が表示される。

【0004】

ここで、「何もない領域」とは、デスクトップ上またはウィンドウ上において、アイコン、メニューバー、ボタンバー、スクロールバー等の起動可能または操作可能なオブジェクトのない領域のことをいう。「操作可能なオブジェクト」とは、デスクトップ上または特定のアプリケーションプログラムのウィンドウ上の操作可能なオブジェクトをいい、アイコン、メニューバー、ボタンバー、スクロールバー等を含むものである。

【0005】

ユーザは、マウスまたはキーボードのスクロールキーを操作して、マウス・カーソル 320 をメニュー内の項目（コマンドを含む。以下、同様）まで移動させ、マウスの左クリックを押すと、当該項目に対応したコマンドを実行することができる。したがって、ユーザはキーボードからコマンドを入力せずすみ、当該コマンドを知らなくても、簡単な操作で入力できるため、この GUI はウィンドウ化された環境で、幅広く提供されている。

【0006】

第4図は、第2の先行技術（特開平 10-333860 号公報）の GUI を示す。同技術は、ユーザがポップアップメニューの表示項目を自由に設定・変更しうる環境を提供する。したがって、第2の先行技術では、使用頻度の高い項目をポップアップメニュー内の上位の位置に設定すれば、当該項目を実行する場合、マウスの操作距離やキーボードのスクロールキーの打鍵回数を節約することができる。第4図は、第3図に示すポップアップメニューにおいて、第2の先行技術により、項目群 331 と項目群 333 の順序を入れ替え、さらに、項目群 333 内の項目「copy」と項目「cut」の順序を入れ替えた場合のポップアップメニューを示す。したがって、ユーザは、ポップアップメニューの上位に位置した項目「copy」に、マウスの移動操作またはスクロールキーの打鍵回数を少なくして、アクセスすることができる。

【0007】

第5図は、第3の先行技術（特開平 5-298049 号公報）の GUI を表す

。同技術によれば、メニューの大きさを限定し、メニュー項目が多すぎて、当該メニューに表示しきれない場合、メニューの右端にスクロール・バー 3 5 0 を提供することができる。したがって、ユーザは、メニュー・バー 3 4 0 の「e d i t」配下のプルダウンメニューの項目が多い場合であっても、メニューにより、背後のウインドウの表示内容が隠されることはなく、編集を効率的に行うことができる。なお、この場合、ユーザは、表示しきれない項目を選択する場合は、マウス等により、スクロール・バー 3 5 0 を操作して、選択する項目を表示させ、当該項目を選択させる必要がある。

【0 0 0 8】

【発明が解決しようとする課題】

しかし、上記先行技術には、いずれにも、ポップアップメニューの下位に位置する項目を選択する場合、入力デバイスの操作量（マウスの物理的な移動量のみならず、マウス・ホイールの操作量、トラックパネルの操作量、トラックポイントの操作量、スクロールキーの打鍵回数も含む）が多くなるため、ユーザの思考が中断されるという問題があった。この問題は、メニューに表示する項目が多くなればなる程、深刻な問題となる。特に、ワードプロセッシングやペインティングに関するソフトウェアにおいては、この問題により、効率的な編集が妨げられることになる。

【0 0 0 9】

そこで、本発明は、ユーザの思考を中断しない、改良されたメニューを提供することを目的とする。

【0 0 1 0】

【課題を解決するための手段】

本発明は、メニューに表示する項目をグルーピングして管理する管理手段と、入力デバイスの特定の操作回数または特定の操作の継続時間に基づき、前記グルーピングしたメニュー項目を切り替えて表示させる表示制御手段とから構成することができる。

【0 0 1 1】

また、他の本発明は、メニューに表示する項目をグルーピングして管理する管



理手段と、オペレーティングシステムから入力デバイスの操作による特定のイベントが発生した旨のメッセージを受信する受信手段と、当該イベントの発生回数に基づき、前記グルーピングしたメニュー項目を切り替えて表示させる表示制御手段とから構成することができる。

## 【 0 0 1 2 】

さらに、他の本発明は、メニューに表示する項目をグルーピングして管理する管理手段と、オペレーティングシステムから入力デバイスの操作による特定のイベントが発生した旨のメッセージを受信する受信手段と、当該イベントの継続時間に基づき、前記グルーピングしたメニュー項目を切り替えて表示させる表示制御手段とから構成することができる。

## 【 0 0 1 3 】

また、本発明の好適な実施の形態においては、上記のグルーピングは、表示するメニュー項目の機能毎にグルーピングされていることが好ましい。他の好適な実施の形態においては、前記表示制御手段は、一定時間以内に、入力デバイスの操作が同一点で操作された場合に起動されることが好ましい。さらに、他の好適な実施の形態においては、前記表示制御手段は、一定時間以内に、入力デバイスの操作が一定の範囲内で操作された場合に起動されることが好ましい。

## 【 0 0 1 4 】

さらに、他の本発明は、上記コンピュータシステムを動作させるためのプログラムを格納したコンピュータ読取り可能な記憶媒体が提供される。より具体的には、他の本発明は、コンピュータのオペレーティングシステムから入力デバイスの操作による特定のイベントが発生した旨のメッセージを受信するプログラムコード手段と、当該イベントの発生回数に基づき、コンピュータシステム上に生成されるグルーピングされたメニュー項目を切り替えて表示させるプログラムコード手段とを有するプログラムを格納したコンピュータ読取り可能な記憶媒体を提供する。さらに、他の本発明は、コンピュータのオペレーティングシステムから入力デバイスの操作による特定のイベントが発生した旨のメッセージを受信するプログラムコード手段と、当該イベントの継続時間に基づき、コンピュータシステム上に生成されるグルーピングされたメニュー項目を切り替えて表示させるプ

ログラムコード手段とを有するプログラムを格納したコンピュータ読取り可能な記憶媒体を提供する。

【0015】

【発明の実施の形態】

以下に、図面を用いて、本発明の好適な実施の形態を説明する。一の図面において、符号により特定されるものは、他の図面において同一の符号により特定されるものと同一の対象を表す。

【0016】

本発明は、主にソフトウェアにより実現されるものであり、C言語等の種々のプログラム言語により記述することができるが、C++、Smalltalk等、その他のオブジェクト指向のプログラム言語により記述するのが望ましい。この明細書では、本発明を実現したソフトウェアを本件ソフトウェアという。本件ソフトウェアは、グラフィカル・ユーザ・インターフェース（GUI）を提供する種々のオペレーティング・システム（OS）上で動作することができる。例えば、そのようなオペレーティング・システムとしては、Microsoft社のWindowsシステム（WindowsはMicrosoft社の登録商標である）、X OpenリミテッドのXWindowシステム（XWindowは、X Openリミテッドの登録商標である）がある。

【0017】

また、本件ソフトウェアは、オペレーティングシステム内でインプリメントすることができる。この場合は、デスクトップ上の何もない領域／オブジェクト上で、入力デバイスの特定の操作したときに、後述するメニューが表示される（第6図、第7図）。また、本件ソフトウェアは、アプリケーションプログラム内でもインプリメントすることができる。この場合は、当該アプリケーションプログラムのウインドウ上の何もない領域／オブジェクト上で、入力デバイスの特定の操作をしたときに、後述するメニューが表示される（第6図、第7図）。ここで、「何もない領域／オブジェクト」とは、前述した定義が適用される。

【0018】

本明細書において、「入力デバイス」とは、ポインティングデバイスやキーボー

ド等のコンピュータシステムに、何らかの入力をするデバイスをいう。ポインティングデバイスには、マウス、ホイール付きマウスのほか、トラックパネル、トラックポイント、トラックボールスタイラスペン等が含まれる。また、キーボードという場合には、キーボード上のスクロールキーや各種英数字キーをも含むものととする。

## 【0019】

また、本明細書において、「イベント」とは、特に断らない限り、入力デバイスの特定の操作により発生する事象をいい、具体的には、マウスの右ボタンアップ、右ボタндаウン、左ボタンアップ、左ボタндаウンのほか、メニュー表示キーの押下、特定のキーの押下、スタイラスペンによるタップ操作により発生する事象をいう。なお、本明細書において、「メニュー」には、ポップアップメニューの他、プルダウンメニュー、プルアップメニュー等も含まれ、また、多階層メニューの場合のカスケードメニューも含まれる。したがって、本件ソフトウェアをメニューバー上の操作に適用した場合、改良されたプルダウンメニューまたはプルアップメニュー（第6図、第7図）を表示させることができる。

## 【0020】

第1図は、本件ソフトウェアが実施されるコンピュータシステムを表したものである。同図中、101は本件ソフトウェアが実施されるコンピュータシステムを、102は本件ソフトウェアのプログラムコード（ソースコード）が記憶された記憶媒体を、103はインターネット等のネットワークを、104は、本件ソフトウェアのソースコードが記憶されたインターネット・プロバイダのサーバの記憶装置（例えば、ハードディスク装置）を表す。なお、コンピュータシステム101は、デスクトップコンピュータを図示したが、ワークステーション、ホストコンピュータ、ノート型コンピュータ、パームトップコンピュータ、パームサイズコンピュータのいずれであってもよい。また、記憶媒体102には、例えば、CD-ROM、光磁気ディスク（MO）、フロッピー・ディスク（FD）の他、搬送波（キャリア・ウェイブ）や通信回線も含まれる。

## 【0021】

第2図は、コンピュータシステム101のハードウェア構成を示したものである。

る。同図中、201は中央処理装置（CPU）を、202は記憶装置（例えば、ハードディスク装置）を、203はメモリを、204は周辺機器、205はキーボード、マウス、トラック・ポイント、トラック・パネル、スタイラスペン等の入力デバイスを、206はディスプレイ等の出力装置を示す。なお、ディスプレイ206には、タッチセンサ付きディスプレイも含み、周辺機器204には、CD-ROMドライブ、MOドライブ、FDドライブ、プリンタ、モデム等が含まれる。

【0022】

記憶媒体102であるCD-ROMに記憶された本件ソフトウェアのソースコードは、コンピュータシステム101のCD-ROMドライブ204等を介して、コンピュータシステム101のハードディスク装置202にインストールされる。本件ソフトウェアの実行時には、コンピュータシステム101のメモリ203にロードされ、CPU201により実行される。

【0023】

また、インターネット・プロバイダのサーバのハードディスク装置104に格納された本件ソフトウェアは、ネットワーク103を介して、コンピュータシステム101にインストールすることができる。この場合においても、本件ソフトウェアのソースコードは、実行時には、コンピュータシステム101のメモリ203にロードされ、CPU201により実行される。

【0024】

第6図は、本ソフトウェアの第1のグラフィカル・ユーザ・インターフェースを表す。同図において明示していないが、マウス・カーソル320のポイントしている箇所は、いずれも、デスクトップ上またはアプリケーションプログラムのウィンドウ上の何もない領域／オブジェクトである。さらに具体的には、マウス・カーソル320のポイントする箇所は、本件ソフトウェアをオペレーティングシステム内でインプリメントする場合は、デスクトップ上の何もない領域／オブジェクトであり、本件ソフトウェアをアプリケーションプログラムでインプリメントする場合は、当該アプリケーションのウィンドウ上の何もない領域／オブジェクトである。

## 【0025】

第6A図は、領域／オブジェクト上で、イベントが1回発生（例えば、マウス・カーソル320を1回クリック）した場合に表示されるポップアップメニュー610を示している。ポップアップメニュー610には、undo（元に戻す）、repeat（繰り返す）、cut（切り取り）、copy（複写）、paste（貼り付け）、delete（削除）等のテキストを編集する項目が機能毎にグルーピングされている。なお、項目undoの横の印611は、undoの配下に、さらに下位のメニューがあることを示している。このように、本件ソフトウェアにより表示されるポップアップメニューには、多階層のメニューを含むことができる。

## 【0026】

第6B図は、領域／オブジェクト上で、イベントが2回発生（例えば、マウス・カーソル320を2回クリック）した場合に表示されるポップアップメニュー620を示している。なお、第8図の説明で後述されるように、この2回目のクリックされたポイントは、1回目のクリックされたポイントと同一の座標であってもよいし、1回目クリックされたポイントからD（設定値）だけ離れた領域内であってもよい。ポップアップメニュー620には、ポップアップメニュー610と異なり、underline（下線付け）、change font（書体の変更）、change size（文字サイズの変更）、insert figure（図形の挿入）、insert graph（グラフの挿入）等のテキストや図形を修飾する項目が機能毎にグルーピングされている。

## 【0027】

第6C図は、領域／オブジェクト上で、イベントが3回発生（例えば、マウス・カーソル320を3回クリック）した場合に表示されるポップアップメニュー630を示している。なお、第8図の説明で後述されるように、この3回目のクリックされたポイントは、1回目のクリックされたポイントと同一の座標であるとしてもできるし、1回目クリックされたポイントから距離D（設定値）だけ離れた領域内であるとしてもできる。ポップアップメニュー630には、ポップアップメニュー610、620と異なり、new file（新ファイルの作成）、open（ファイルを開く）、close（ファイルを閉じる）、save（上書き保存）、

property (ファイルの属性), exit (アプリケーションプログラムの終了) 等のファイルを操作する項目が機能毎にグルーピングされている。

## 【0028】

この第6図の説明から理解されるように、入力デバイスの特定の操作回数に基づき、グルーピングされた項目を、ポップアップメニューに切り替えて表示するため、ユーザは、マウスの移動量または特定キーの打鍵回数を少なくして、ポップアップメニュー内の項目を選択することができる。また、本件ソフトウェアにおいては、グルーピングされた項目は、必ずしも機能毎にグルーピングする必要はないが、第6図において説明されたように、機能毎にグルーピングされる方が、ユーザ・フレンドリーなグラフィカル・ユーザ・インターフェースを提供する点で、望ましい。また、第15図の説明において後述するように、切り替える順番は、ユーザが任意に設定、変更することができる。

## 【0029】

さらに、定義されたメニューの数(N)より多い回数(N+1以上)だけ操作された場合に、第1番目のメニューに戻って、メニューを表示させてもよいし、第N番目のメニューを表示し続けてもよい。

## 【0030】

第7図は、本ソフトウェアの第2のグラフィカル・ユーザ・インターフェースを表す。同図において明示していないが、マウス・カーソル320のポイントしている箇所は、領域/オブジェクトである。さらに具体的には、第6図の場合と同様に、マウス・カーソル320のポイントする箇所は、本件ソフトウェアをオペレーティングシステム内でインプリメントする場合は、デスクトップ上の何もない領域/オブジェクトであり、本件ソフトウェアをアプリケーションプログラムでインプリメントする場合は、当該アプリケーションのウィンドウ上の何もない領域/オブジェクトである。

## 【0031】

第7A図は、イベントがT(設定値)秒間継続した場合(例えば、領域/オブジェクト上で、マウス・カーソル320をT秒間継続してクリックした場合)に表示されるポップアップメニュー710を示している。ここでは、説明の簡単な

ため、ポップアップメニュー 710 に表示される項目は、第 6 図のポップアップメニュー 610 に表示される項目と同じであり、テキストを編集する項目が機能毎にグルーピングされている。なお、ポインティングデバイスの特定の操作の継続時間 T は、ユーザが任意に設定することができ、例えば、0.1 秒、0.5 秒、1 秒、1.5 秒等を設定することができる。

#### 【0032】

第 7 B 図は、イベントが 2 T 秒間継続して発生した場合（例えば、領域／オブジェクト上で、マウス・カーソル 320 を、2 T 秒間継続してクリックした場合）に表示されるポップアップメニュー 720 を示している。なお、第 8 図の説明で後述されるように、継続してクリックされるポイントは、1 回目のクリックされたポイントと同一の座標であってもよいし、1 回目クリックされたポイントから D（設定値）だけ離れた領域内であってもよい。また、説明の簡単のため、ポップアップメニュー 720 に表示されている項目は、ポップアップメニュー 620 に表示されている項目と同じであり、テキストを修飾する項目が機能毎にグルーピングされている。

#### 【0033】

第 7 C 図は、イベントが 3 T 秒間継続して発生した場合（領域／オブジェクト上で、マウス・カーソル 320 を、3 T 秒間継続してクリックした場合）に表示されるポップアップメニュー 730 を示している。なお、第 8 図の説明で後述されるように、継続してクリックされるポイントは、1 回目のクリックされたポイントと同一の座標であってもよいし、1 回目クリックされたポイントから D（設定値）だけ離れた領域内であってもよい。また、説明の簡単のため、ポップアップメニュー 730 に表示されている項目は、ポップアップメニュー 630 に表示されている項目と同じであり、ファイル进行操作する項目が機能毎にグルーピングされている。

#### 【0034】

この第 7 図の説明から理解されるように、入力デバイスの特定の操作の継続時間に基づき、グルーピングされた項目を、ポップアップメニューに切り替えて表示するため、ユーザは、マウスの移動量または特定キーの打鍵回数を少なくして

、ポップアップメニュー内の項目を選択することができる。また、本件ソフトウェアにおいては、グルーピングされた項目は、必ずしも機能毎にグルーピングする必要はないが、第6図において説明されたように、機能毎にグルーピングされる方が望ましい。また、このGUIにおいて、ユーザは、ポップアップメニューを切り替える時間を、設定・変更することができ、また、切り替える順番も設定・変更することができる。

## 【0035】

さらに、この場合においても、定義されたメニューの数(N)より多い時間( $(N+1)T$ 秒以上)だけ継続して操作された場合に、第1番目のメニューに戻って、メニューを表示させてもよいし、第N番目のメニューを表示し続けてもよい。

## 【0036】

第8図は、本件ソフトウェアにおけるポップアップメニューの切り換えの条件を示すものである。以下に説明する条件は、本発明の完成に必須のものではなく、設定されることが望ましいものである。

## 【0037】

第8Aの条件は、第2回目以降(第6図)にポインティングデバイスの操作がされたマウス・カーソルの座標または $2T$ 秒以降(第7図)に継続して操作されるマウス・カーソルの座標が、第1回目のポインティングデバイスの特定の操作が行われたマウス・カーソルの座標( $X_0$ ,  $Y_0$ )と同一点であることを要求する。したがって、ユーザは、このような厳しく制限された条件の下で、本件ソフトウェアによるポップアップメニューの切り換えを実行できるため、頻繁にポップアップメニューの切り換えが必要ない場合やポップアップメニューの第1頁目(第6A図、第7A図)に頻繁に選択する項目があり、第2頁目以降のポップアップメニューをあまり使用しない場合に有効な条件である。なお、( $X_0$ ,  $Y_0$ )は、デスクトップ上の座標とすることができ、また、特定のアプリケーションプログラムのウィンドウ内の座標とすることができる。

## 【0038】

第8B図の条件は、第2回目以降(第6図)にポインティングデバイスの操作



がされたマウス・カーソルの座標または2 T秒以降（第7図）に継続して操作されるマウス・カーソルの座標が、第1回目のポインティングデバイスの特定の操作が行われたマウス・カーソルの座標（X<sub>0</sub>, Y<sub>0</sub>）から距離Dの範囲内にあることを許容する条件である。距離Dは、ユーザにより、任意に設定・変更可能である。したがって、ユーザは、このような緩やかな条件の下で、本件ソフトウェアによるポップアップメニューの切り換えを実行できるため、頻繁にポップアップメニューの切り換えが必要な場合や落ち着きのないユーザが使用する場合に有効な条件である。

## 【0039】

なお、この場合、距離D以内の領域内であっても、座標（X<sub>0</sub>, Y<sub>0</sub>）の属するオブジェクトと座標（X, Y）の属するオブジェクトが異なる場合は、回数を表すカウンタや継続時間を表すタイマをリセットしてもよい。具体的には、距離D以内の領域内であっても、座標（X<sub>0</sub>, Y<sub>0</sub>）がオブジェクトY上にあり、座標（X, Y）が他のオブジェクトZまたはデスクトップ領域にある場合は、オブジェクトYのカウンタやタイマをリセットして、座標（X, Y）の属するアプリケーションプログラム（オブジェクトZ）で、カウンタまたはタイマを起動し、当該アプリケーションプログラムで提供されるメニューの切り換えを、新たに行うことができる。

## 【0040】

第9図は、OSであるMS Windows上で動作するアプリケーションプログラムで、本件ソフトウェアをインプリメントした場合の、OSとアプリケーションプログラム間のメッセージのやり取りの具体例を説明する図である。なお、本件ソフトウェアをOSでインプリメントする場合は、メッセージ送信がOS内に閉じる点を除けば、アプリケーションプログラムでインプリメントする場合と、基本的に同様である。したがって、以下の説明では、本件ソフトウェアをアプリケーションプログラムでインプリメントする場合の動作等を説明し、OSでインプリメントする場合の説明は省略する。

## 【0041】

第9図において、910はマウス、キーボード等の入力デバイスを、920は

オペレーティングシステムを、930は本件ソフトウェアを示す。まず、ユーザは、入力デバイス910の特定の操作（例えばマウスの右クリック、特定キーの押下）を行う。OS920のデバイスドライバは、この特定の操作によるイベントが発生したことを検出する。本件ソフトウェア930は、その起動時に、OS920に対して、このイベントのモニタを依頼している。このため、OS920は、本件ソフトウェア930のメッセージキュー923に当該イベントが発生したことを知らせるメッセージを、本件ソフトウェア930のメッセージ・キュー923にキューイングし、メッセージを、本件ソフトウェア930に送信する。MSWindowsの場合、このメッセージは、当該イベントに対応した変数に格納される。

## 【0042】

具体的には、イベントがマウスの右ボタンアップであれば、当該変数はWM\_BUTTONUPであり、イベントがマウスの右ボタンダウンであれば、当該変数はWM\_BUTTONDOWNである。また、イベントがマウスの左ボタンアップであれば、当該変数はWM\_LBUTTONDOWNであり、イベントがマウスの左ボタンダウンであれば、当該変数はWM\_LBUTTONDOWNである。また、イベントがメニュー表示キーの押下であれば、当該変数はWM\_CONTEXTMENUである。また、イベントが通常のキーダウンであれば、当該変数はWM\_KEYDOWNであり、イベントが通常のキーアップであれば、当該変数はWM\_KEYUPである。さらに、イベントが特定キーの押下である場合、当該変数は当該特定キーのキャラクタコードを格納するWM\_CHARである。以下では、これらの各イベントに対応したメッセージを、まとめてWM\_XXXと書く。

## 【0043】

本件ソフトウェア930のメッセージループ931は、メッセージWM\_XXXを受信すると、そのメッセージ内容を判定し、その判定結果に応じた処理を行う。この実施の形態では、メッセージループ931が、本発明の受信手段に相当する。この場合の処理は、第8図を用いて説明したような条件が設定されていれば、その条件を満たすか否かの判定を行う。当該条件を満たしていると判定した場合は、ウインドウ・プロシージャ933またはウインドウ・プロシージャ933か

ら呼び出されるルーチン 9 3 5 は、リソースファイルの内容（例えば、第 1 6 図）を、デスクトップ上に、メニューとしてを表示させることができる。

【 0 0 4 4 】

なお、第 8 A 図、第 8 B 図の条件判定には、デスクトップ上またはウインドウ上におけるイベント発生時のマウスカーソルの座標が必要となるが、この実施の形態においては、当該座標は、メッセージ WM\_MOUSEMOVE と共に、引数 lParam 内に格納されて、アプリケーションプログラムに送信される。メッセージ WM\_MOUSEMOVE は、マウス・カーソルが移動した時に送られるメッセージであり、その引数 lParam には、マウスカーソルの x 座標 (xPos)、マウスカーソルの y 座標 (yPos) を含む。メッセージ WM\_MOUSEMOVE は、マウスがキャプチャされているときは、マウスをキャプチャしているウインドウのウインドウ・プロシージャ 9 3 3 に、マウスがキャプチャされていない場合は、マウスカーソルの下にあるウインドウのウインドウ・プロシージャ 9 3 3 に送られる。

【 0 0 4 5 】

また、(revised) この実施の形態では、リソースファイル（例えば、第 1 6 図）が、本発明の管理手段に相当する。この場合、メニューの切り換えは、API 関数 DestroyMenu を用いて、現在表示しているメニューを破棄し、API 関数 TrackPopupMenu を用いて、次のメニューを定義するリソースファイルの内容を表示する。また、この実施の形態では、ウインドウ・プロシージャ 9 3 3 またはウインドウ・プロシージャ 9 3 3 から呼び出されるルーチン 9 3 5 が表示制御手段に相当する。

【 0 0 4 6 】

また、OS 9 2 0 で提供される API 関数 9 2 5（例えば、CreatePopupMenu）を使用していれば、第 9 図に示すように、ウインドウ・プロシージャ 9 3 3 は、API 関数 CreatePopupMenu を呼び出して、メニューを作成し、メニューの切り換えを行う。この場合、メニューの切り換えは、API 関数 DestroyMenu を用いて、現在表示しているメニューを破棄し、API 関数 CreatePopupMenu を用いて、次のメニューを作成し、API 関数 TrackPopupMenu を用いて、当該ファイルの内容をメニューとして表示する。なお、API 関数 CreatePopupMenu を使用する場合、本発明の管

理手段として、第17図に例示されるようなファイルを用いることができる。この第17図に例示されるファイルは、リソースファイルの形式に従っていない点で、リソースファイルとは異なり、その形式は、後述するように、アプリケーションプログラムで自由に定義することができる。また、この実施の形態においても、ウインドウ・プロシージャ933またはウインドウ・プロシージャ933から呼び出されるルーチン935が表示制御手段に相当する。

【0047】

第10図は、本件ソフトウェア930の動作の概要を示すフローチャートである。まず、ステップ1010において、ユーザは、デスクトップ上で、アイコンの左クリックまたはメニューの選択により、本件ソフトウェア930を起動（活性化）する。

【0048】

次に、ステップ1020において、本件ソフトウェア930は、OSに対して、特定のイベントの監視を依頼する。この場合、特定のイベントとは、入力デバイスの特定の操作がされたというイベントをいい、例えば、マウスの右クリック、左クリック、特定キーの押下が含まれる。

【0049】

次に、ステップ1030において、本件ソフトウェア930のメッセージループ931は、OSから、当該イベントが発生した旨のメッセージを受信したか否かを判定する。

【0050】

具体的には、メッセージWM\_XXXの値が所定値であるか否かを判定する。メッセージループ931は、受信したメッセージWM\_XXXが所定値、すなわち、前記イベントが発生したことを示す値を取るまで、この判定を繰り返す。メッセージループ931は、受信したメッセージWM\_XXXが所定値を取ると判定したときは、ステップ1140に進む。

【0051】

次に、ステップ1040において、ウインドウ・プロシージャ933は、例えば第8図で説明したような、所定の条件が設定されている場合は、各イベントが

その条件を具備しているか否かを判定する。具備していない場合は、ステップ1030に進み、具備している場合は、ステップ1050に進む。なお、この条件が設定されていない場合は、ステップ1040は省略することが可能であり、この条件設定は必ずしも必要ではない。

【0052】

次に、ステップ1050において、本件ソフトウェア930は、イベント発生回数またはイベントの継続発生時間に基づき、第6図または第7図に例示したように、メニューを切り替えてデスクトップ上に表示する。

【0053】

第11図は、第8A図に対応した条件を設定した場合の本件ソフトウェア930のフローを示す。これは、第10図のステップ1030からステップ1040を詳述したフローである。

【0054】

ステップ1110において、本件ソフトウェア930のメッセージループ931は、監視依頼したイベントが発生した旨のメッセージWM\_XXXを受信したか否かを判定する。メッセージループ931は、このメッセージを受信しなければ、当該判定を継続する。一方、メッセージループ931は、このメッセージを受信し、その値がイベントが発生したことを示す値と判断すると、ウインドウ・プロシージャ933は、所定のルーチン935を呼び出し、ステップ1120に進む。この場合、ウインドウ・プロシージャ933は、当該メッセージと共に、イベントの発生した時の、デスクトップ上におけるマウス・カーソルの座標(X, Y)を取得する。

【0055】

次に、ステップ1120において、当該ルーチン935は、基準となる中心座標(X<sub>o</sub>, Y<sub>o</sub>)として、ステップ1110で取得した座標(X, Y)を設定する。すなわち、基準となる中心座標(X<sub>o</sub>, Y<sub>o</sub>)には、非継続的にイベントが生じる場合の1回目または継続的にイベントが生じる場合の最初の座標が設定される。次に、当該ルーチン935は、ステップ1130において、連続して、または継続して行われるイベントを監視するためのタイマをスタートさせる。なお

、ステップ 1140 については、第 13 図、第 14 図のフローの説明で後述する。

【0056】

次に、ステップ 1150 において、当該ルーチン 935 は、イベント発生時の座標 (X, Y) が中心座標 (X<sub>o</sub>, Y<sub>o</sub>) と等しいか否かを判定する。両座標が等しくないと判定した場合は、当該ルーチン 935 は、ステップ 1160 に進み、何もしない。一方、両座標が等しいと判定した場合は、ステップ 1170 へと進む (第 13 図 1310、第 14 図 1410 に続く)。なお、現在は、1 回目 (J=1) の処理なので、この条件は満たされ、ステップ 1170 へと進む。

【0057】

第 12 図は、第 8 B 図に対応した条件を設定した場合の本件ソフトウェア 930 のフローを示す。これも、第 11 図同様、第 10 図のステップ 1030 からステップ 1040 を詳述したフローである。

【0058】

ステップ 1210 において、本件ソフトウェア 930 のメッセージループ 931 は、監視依頼したイベントが発生した旨のメッセージ WM\_XXX を受信したか否かを判定する。メッセージループ 931 は、このメッセージを受信しなければ、当該判定を継続する。一方、メッセージループ 931 は、このメッセージを受信し、その値がイベントの発生を表す値と判定すると、ウインドウ・プロシージャ 933 は、所定のルーチン 935 を呼び出し、ステップ 1220 に進む。この場合、ウインドウ・プロシージャ 933 は、当該メッセージと共に、イベントの発生した時の、デスクトップ上におけるマウス・カーソルの座標 (X, Y) を取得する。

【0059】

次に、ステップ 1220 において、当該ルーチン 935 は、基準となる中心座標 (X<sub>o</sub>, Y<sub>o</sub>) として、ステップ 1210 で取得した座標 (X, Y) を設定する。この場合も、基準となる中心座標 (X<sub>o</sub>, Y<sub>o</sub>) には、非継続的にイベントが生じる場合の 1 回目または継続的にイベントが生じる場合の最初の座標が設定される。次に、当該ルーチン 935 は、ステップ 1230 において、連続して、

または継続して行われるイベントを監視するためのタイマをスタートさせる。なお、ステップ1240については、第13図、第14図のフローの説明で後述する。

#### 【0060】

次に、ステップ1250において、当該ルーチン935は、イベント発生時の座標(X, Y)が中心座標(X<sub>o</sub>, Y<sub>o</sub>)から距離D以内にあるか否かを判定する。ここで、距離Dは、ユーザまたはアプリケーションプログラムにおいて、任意に設定・変更が可能であり、例えば、第15図の画面を用いてされる。ここで、座標(X, Y)と中心座標(X<sub>o</sub>, Y<sub>o</sub>)の距離dは、 $d = [(X - X_o)^2 + (Y - Y_o)^2]^{1/2}$ で定義される。

#### 【0061】

距離dが設定値Dより大きいと判定した場合は、当該ルーチン935は、ステップ1260に進み、何もしない。一方、距離dが設定値Dより小さいか等しいと判定した場合は、ステップ1270へと進む(第13図1310、第14図1410に続く)。

#### 【0062】

第13図、第14図は、第10図のステップ1050(メニューの切り換え)を詳述したものである。第13図は、メニューの切り換えを、非連続的に発生するイベントの回数に基づいて行う場合のフローを示す。ステップ1310は、第11図のステップ1170または第12図ステップ1270から続くものである。

#### 【0063】

次に、ステップ1320において、前記ルーチン935は、発生したイベントの回数を示すカウンタJを、1だけカウントアップする(J = J + 1)。カウンタJの初期値は0に設定されているため、現在、Jの値は1である。

#### 【0064】

次に、ステップ1330において、当該ルーチン935は、カウンタJと関連づけられて定義されたリソースファイル(例えば、第16図)を、デスクトップ上に表示させる。

## 【0065】

ステップ1340において、当該ルーチン935は、ポップアップメニュー（J）内の項目が選択されたかを判定する。この場合も、ルーチン935は、メッセージループ931により、オペレーティングシステムから送信されるその旨のメッセージの値を用いて判定する。ポップアップメニュー（J）内の項目が選択されたと判定した場合は、ステップ1350に進み、ウインドウ・プロシージャ933は、選択された項目に応じた処理を行う。一方、ポップアップメニュー（J）内の項目が選択された旨のメッセージを受信しない場合は、ステップ1360において、当該ルーチン935は、イベントが発生した旨のメッセージを受信したか否かを判定する。イベントが発生した旨のメッセージを受信した場合は、ルーチン935は、ステップ1130において起動したタイマtを用いて、当該イベントが一定時間以内か否かにより制限をつけることができる。したがって、当該イベントが発生した場合であっても、一定時間が経過した後は、当該イベントの発生を無視することも可能である。

## 【0066】

当該ルーチン935は、一定時間以内にイベントが発生した旨のメッセージを受信しない場合は、ステップ1370に進み、何も処理を行わない。一方、一定時間以内にイベントが発生した旨のメッセージを受信した場合は、ステップ1380に進み、さらに、第11図のステップ1140または第12図のステップ1240に進む。この場合も、メッセージWM\_MOUSEMOVEと共に、マウスカーソルの座標（X，Y）を、OSから取得している。

## 【0067】

第11図ステップ1150または第12図ステップ1250において、ウインドウ・プロシージャ933が所定の条件を満たすか否かを判定する。具体的には、第8A図の場合、2回目にイベントが発生したときの座標（X，Y）が、中心座標（X<sub>o</sub>，Y<sub>o</sub>）（1回目にイベントが発生したときの座標）と等しいか否かを判定する。第8B図の場合、2回目にイベントが発生したときの座標（X，Y）が、中心座標（X<sub>o</sub>，Y<sub>o</sub>）から距離D以内にあるか否かを判定する。これらの条件を満たす場合は、ルーチン935は、カウンタJをカウントアップして（



ステップ1320)、Jを2に更新し、J=2に対応したポップアップメニューをデスクトップ上に表示する(ステップ1340)。

【0068】

J=3以降の処理も同様である。この場合、中心座標( $X_o$ ,  $Y_o$ )を、カウンタJの更新と共に、更新していてもよい。具体的には、J番目の中心座標( $X_oj$ ,  $Y_oj$ )を、(J-1)番目のイベントの発生した時のマウスカーソルの座標(X, Y)として、カウンタJと共に更新することもできる。

【0069】

第14図は、メニューの切り換えを、連続的に発生するイベントの継続時間に基づいて行う場合のフローを示す。ステップ1410は、第11図のステップ1170または第12図ステップ1270から続くものである。

【0070】

次に、ステップ1420において、前記ルーチン935は、発生したイベントの継続時間の指標を示すカウンタnを、1だけカウントアップする( $n=n+1$ )。カウンタnの初期値は0に設定されているため、現在、nの値は1である。

【0071】

次に、ステップ1430において、当該ルーチン935は、タイマtとnTとの大小関係を判定する。ここで、Tはポップアップメニューの単位切り換え時間を示すパラメータである。パラメータTは、アプリケーションプログラムにより設定することができ、また、ユーザにより設定・変更が可能である。したがって、 $T=1$ (秒)と設定されている場合は、イベントが1秒継続する度に、ポップアップメニューが切り替えて表示される。

【0072】

タイマtがnTより小さいと判定された場合、ルーチン935は、ステップ1430に進み、イベントが継続しているかを判定する。イベントが継続していない、すなわち、イベントが中断されたと判断した場合は、当該ルーチン935は、ステップ1440に進み、何もしない。一方、タイマtがnTと等しいまたは大きいと判定された場合、当該ルーチン935は、ステップ1450に進み、カウンタnと関連づけられて定義されたリソースファイル(例えば、第16図)を

、デスクトップ上に表示させる。現在、 $n = 1$ なので、第6A図または第7A図に示したようなポップアップメニューがデスクトップ上に表示される。

#### 【0073】

次に、当該ルーチン935は、ステップ1460に進み、ポップアップメニュー（ $n$ ）内の項目が選択されたかを判定する。当該ルーチン935は、オペレーティングシステムから送信されるその旨のメッセージの値を用いて判定する。ポップアップメニュー（ $n$ ）内の項目が選択されたと判定した場合は、ステップ1470に進み、当該ルーチン935は、選択された項目に応じた処理を行う。一方、ポップアップメニュー（ $n$ ）内の項目が選択された旨のメッセージを受信しない場合は、ステップ1480において、当該ルーチン935は、イベントが継続しているか否かを判定する。イベントが継続していると判定した場合は、当該ルーチン935は、ステップ1495に進み、さらに、第11図のステップ1140または第12図のステップ1240に進む。一方、イベントが継続していないと判断した場合は、ステップ1490に進み、何もしない。なお、この場合も、メッセージWM\_MOUSEMOVEと共に、マウスカーソルの座標（ $X$ ， $Y$ ）を、OSから取得している。

#### 【0074】

次に、ステップ1495に進んだ場合、ルーチン935は、第11図ステップ1150または第12図ステップ1250において、所定の条件を満たすか否かを判定する。具体的には、第8A図の場合、2回目にイベントが発生したときの座標（ $X$ ， $Y$ ）が、中心座標（ $X_0$ ， $Y_0$ ）（1回目にイベントが発生したときの座標）と等しいか否かを判定する。第8B図の場合、2回目にイベントが発生したときの座標（ $X$ ， $Y$ ）が、中心座標（ $X_0$ ， $Y_0$ ）から距離 $D$ 以内にあるか否かを判定する。これらの条件を満たす場合は、ルーチン935は、カウンタ $n$ をカウントアップして（ステップ1420）、 $n$ を2に更新する。タイマ $t$ が $T$ と等しいまたは大きいと判定した場合は、当該ルーチン935は、 $n = 2$ と関連づけられたポップメニューを、デスクトップ上に表示する（ステップ1430）。この場合、 $n = 2$ なので、例えば、第6B図または第7B図に示されるポップアップメニューがデスクトップ上に表示される。

## 【0075】

$n=3$ 以降の処理も同様である。この場合、中心座標( $X_o$ ,  $Y_o$ )を、カウンタ $J$ の更新と共に、更新していてもよい。具体的には、 $n$ 番目の中心座標( $X_{on}$ ,  $Y_{on}$ )を、( $n-1$ )番目のイベントの発生した時のマウスカーソルの座標( $X$ ,  $Y$ )として、カウンタ $n$ と共に更新することもできる。

## 【0076】

第15図は、ポップアップメニューを定義する際に用いられるグラフィカル・ユーザ・インターフェースの例を示す。このインターフェースは、ソフトウェア技術者の間では、周知のインターフェースである。第15A図においては、関連づけられるイベントは、マウスの右クリックである(1510)。したがって、この例では、マウスの右クリックの回数またはマウスの右クリックの継続時間に基づいて、ポップアップメニューが切り替えられることになる。

## 【0077】

1520は、項目の機能毎に管理されたポップアップメニューを示す。具体的には、項目1521は編集を示すポップアップメニュー(例えば、第6A図、第7A図)を、項目1522は修飾/図形を示すポップアップメニュー(例えば、第6B図、第7B図)を、項目1523はファイル操作を示すポップアップメニュー(例えば、第6C図、第7C図)を示す。ユーザは、項目1520の内容を、1531から1535に示すボタンを操作することにより、編集することができる。例えば、ユーザは、ボタン1531の押下により、1520に新たな項目を追加することができ、ボタン1532の押下により、1520の項目を編集することができ、ボタン1533の押下により、1520から、項目1521乃至項目1523を削除することができる。さらに、ユーザは、ボタン1534の押下により、ウインドウ1500を破棄(destroy)することができ、ボタン1535の押下により、既に行った処理を取消することができる。

## 【0078】

ユーザは、ウインドウ1500(第15A図)において、各項目1521から1523を選択して、ダブルクリックまたは編集ボタン1532を押下することにより、第15B図に示すウインドウ1540を開くことができる。第15B図

は、ウインドウ 1500 において、編集を示す項目 1521 が選択された場合を示す (1550)。左側のボックス 1560 には、現在、編集を示すポップアップメニューに属する項目が表示される。これは、第 6 A 図または第 7 A 図のポップアップメニューに相当する。ユーザは、ボタン 1581 の押下によりボックス 1560 に新たな項目を追加することができ、ボタン 1582 の押下により、ボックス 1570 の項目を削除することができる。ボックス 1570 は、編集後のポップアップメニューを示す。また、ボタン 1591 の押下により、ユーザは、ウインドウ 1540 における編集内容を設定することができ、ボタン 1592 の押下により、既にした編集内容を取消することができる。

【0079】

第 16 図は、第 15 図の GUI により設定されたリソースファイル (管理手段に相当する) を示す。第 15 図では、機能毎に分類してポップアップメニューを定義した結果、生成されるリソースファイルも機能毎に分類されている。なお、本発明の完成においては、必ずしも、ポップアップメニューは機能毎に分類されている必要はない。

【0080】

第 16 A 図は、第 15 A 図の項目 1521 に対応したリソースファイル 1600 を示している。このリソースファイル 1600 を用いて、第 6 A 図、第 7 A 図のポップアップメニューを表示することができる。インクルード文 1601 は、ヘッダファイル "popup.h" を読み込むものである。BEGIN 文 1610 は、ポップアップメニュー内に表示する項目が始まることを宣言し、END 文 1611 は、その項目が終わることを宣言する。

【0081】

MENUITEM 文 1620、1622 から 1628 は、ポップアップメニューに表示する各項目を定義する。具体的には、MENUITEM 文の後のダブルクォーテーション ( " ) で囲まれた文字列がポップアップメニューに表示される。その項目が選択された場合、MENUITEM 文のカンマ後の変数がメッセージ WM\_COMMAND に格納され、そのメニューを所持するウインドウ・プロシージャ 933 に送信される。したがって、MENUITEM 文 1620 の場合、第 6 A 図、第 7 A 図に示すように、ポッ

ポップアップメニューの第1行目に、“undo”が表示され、項目“undo”が選択された場合、OS 920は、メッセージWM\_COMMANDに、IDM\_UNDOを格納して、本件ソフトウェア930のウィンドウ・プロシージャ933に対して送信する。

## 【0082】

文1623、文1627は、ポップアップメニュー内に、区切り線（セパレータ）を入れることを指定する文であり、ポップアップメニューには、第6A図の“-----”のように表示される。ブロック1621は、項目“undo”の下位のメニューを示す。したがって、ブロック1621は、ポップアップメニューでは、第6A図611のように表示され、項目“undo”が選択されたときには、カスケードメニューとして表示される。このように、ブロック1621を入れ子構造にすることにより、複数の階層のメニューを作成することができる。

## 【0083】

第16B図は、第15A図の項目1522に対応したリソースファイル1650を示している。このリソースファイル1650を用いて、第6B図、第7B図に示すポップアップファイルを表示することができる。なお、リソースファイル1650の説明は、リソースファイル1600（第16A図）の説明と重複するので、省略する。また、第6C図、第7C図に対応したリソースファイルも、同様にして定義できるが、紙面の関係上、明示を省略する。

## 【0084】

第17図は、第15図のGUIにより設定された、管理手段の他の形態を示す。第17図で示すファイルは、API関数CreatePopupMenuを使用する場合に、メニューを定義するファイルの一例である。また、第16図の場合と同様に、第15図では、機能毎に分類してポップアップメニューを定義した結果、第17図で生成されるファイルも機能毎に分類されている。なお、本発明の完成においては、必ずしも、ポップアップメニューは機能毎に分類されている必要はない。

## 【0085】

第17A図は、第15A図の項目1521に対応したファイル1710を示している。このファイル1710を用いて、第6A図、第7A図のメニューを表示することができる。なお、上述したように、これらのファイルは、リソースファ

イルの形式（第 1 6 図）に従う必要はなく、アプリケーションプログラム側で、その形式を定義することができる。ファイル 1 7 1 0 の形式では、ある行の第一のカンマ前の項目は、メニュー内に表示する項目を示し、第一のカンマ後の項目は、第一のカンマ前の項目が選択された場合に、変数 WM\_COMMAND に格納される変数を示している。したがって、ファイル 1 7 1 0 の例では、メニューの 1 行目に表示される項目は "undo" であり、"undo" が選択されると、変数 WM\_COMMAND の値は 0x1001 となる。また、カンマのみの行は、メニュー中にセパレータ "-----" を挿入することを意味し、第 6 A 図、第 7 A 図に示したように表示される。また、ファイル 1 7 1 0 には、図示していないが、多階層を含むメニューの場合は、そのメニューアイテムの下に、下層のメニュー項目を SUB-ITEM1, SUB-ITEM2 と記載したり、BEGIN 文と END 文の間に挟んで定義することができる。

## 【 0 0 8 6 】

第 1 7 B 図は、第 1 5 A 図の項目 1 5 2 2 に対応したファイル 1 7 2 0 を示している。ファイル 1 7 2 0 を用いて、本件ソフトウェアは、第 6 B 図、第 7 B 図に示すメニューを表示することができる。また、第 1 7 C 図は、第 1 5 A 図の項目 1 5 2 3 に対応したファイル 1 7 3 0 を示している。ファイル 1 7 3 0 を用いて、本件ソフトウェアは、第 6 C 図、第 7 C 図に示すメニューを表示することができる。これらのファイルの形式は、第 1 7 A 図で説明した通りである。

## 【 0 0 8 7 】

また、管理手段の他の例として、Microsoft 社の Windows95 または Windows98（Windows は Microsoft 社の登録商標である）で採用しているレジストリを採用することもできる。レジストリを採用した場合は、第 1 6 図、第 1 7 図に例示されるファイルからの読み込み処理が不要となるため、さらにアプリケーションプログラム側の処理が簡単になる。

## 【 0 0 8 8 】

この場合、メニュー項目は、レジストリ内に各階層のメニューを定義し、レジストリ関数を用いて、これらのメニューを呼び出し、表示させる。具体的には、レジストリファイル HKEY\_LOCAL\_MACHINE\_¥¥SOFTWARE¥¥COMPANY\_¥¥POPUP 以下に各階層のメニューを定義する。例えば、レジストリキー HKEY\_LOCAL\_MACHI

NE\_¥¥SOFTWARE¥¥COMPANY\_\_A¥¥POPUP¥¥MENU1で参照されるレジストリファイルに、第 6 A 図、第 7 A 図に対応したメニューを定義する。同様に、レジストリファイルHKEY\_\_LOCAL\_\_MACHINE\_\_¥¥SOFTWARE¥¥COMPANY\_\_A¥¥POPUP¥¥MENU2で参照されるレジストリファイルに、第 6 B 図、第 7 B 図に対応したメニューを定義し、また、レジストリキーHKEY\_\_LOCAL\_\_MACHINE\_\_¥¥SOFTWARE¥¥COMPANY\_\_A¥¥POPUP¥¥MENU3で参照されるレジストリファイルに、第 6 C 図、第 7 C 図に対応したメニューを定義する。

【 0 0 8 9 】

これらのレジストリファイルに定義した各メニュー項目は、レジストリキーにより、その値を参照することができる。これらのキー値を参照する場合には、レジストリ関数を用い、本件ソフトウェアで用いるレジストリ関数は、レジストリファイルをオープンするRegOpenKeyEx()、レジストリキーの値を読み出すRegQueryValueEx()、レジストリファイルをクローズするRegCloseKey()がある。

【 0 0 9 0 】

例えば、MENU1内の 1 行目の項目"undo" (第 6 A 図、第 7 A 図のメニューの 1 行目) は、レジストリキーHKEY\_\_LOCAL\_\_MACHINE\_\_¥¥SOFTWARE¥¥COMPANY\_\_A¥¥POPUP¥¥MENU1¥1¥nameの値("undo")がメニューに表示され、また、その項目が選択された場合は、レジストリキーHKEY\_\_LOCAL\_\_MACHINE\_\_¥¥SOFTWARE¥¥COMPANY\_\_A¥¥POPUP¥¥MENU1¥1¥idの値("0x1001")がWM#\_\_COMMANDに格納される。また、同様に、MENU1内の 2 行目の項目"repeat" (第 6 A 図、第 7 A 図のメニューの 2 行目) は、レジストリキーHKEY\_\_LOCAL\_\_MACHINE\_\_¥¥SOFTWARE¥¥COMPANY\_\_A¥¥POPUP¥¥MENU1¥2¥nameの値("repeat")がメニューに表示され、また、その項目が選択された場合は、レジストリキーHKEY\_\_LOCAL\_\_MACHINE\_\_¥¥SOFTWARE¥¥COMPANY\_\_A¥¥POPUP¥¥MENU1¥2¥idの値("0x1002")がWM#\_\_COMMANDに格納される。

【 0 0 9 1 】

アプリケーションプログラムは、このようなレジストリ関数を用いて、レジストリキーによりレジストリファイルの内容を参照することで、上述したようなメニューの切り換えを実現することができる。

【 0 0 9 2 】

【発明の効果】

以上の説明から明らかなように、本発明によれば、メニューを表示する場合に、入力デバイスの操作量を少なくすることができる。したがって、メニューの操作による、ユーザの思考の中断を軽減することができる。

【図面の簡単な説明】

【図 1】

本件ソフトウェアが実行されるコンピュータシステムの外觀である。

【図 2】

本件ソフトウェアが実行されるコンピュータシステムのハードウェア構成である。

【図 3】

第 1 の先行技術のグラフィカル・ユーザ・インターフェースである。

【図 4】

第 2 の先行技術のグラフィカル・ユーザ・インターフェースである。

【図 5】

第 3 の先行技術のグラフィカル・ユーザ・インターフェースである。

【図 6】

本発明の第 1 のグラフィカル・ユーザ・インターフェースである。

【図 7】

本発明の第 2 のグラフィカル・ユーザ・インターフェースである。

【図 8】

表示制御手段を起動するための条件を説明する図である。

【図 9】

OS - AP 間のメッセージの流れを説明する図である。

【図 1 0】

本件ソフトウェアのフローチャートである。

【図 1 1】

第 8 A 図に示す条件に対応したフローチャートである。

【図 1 2】



第 8 B 図に示す条件に対応したフローチャートである。

【図 1 3】

入力デバイスの特定の操作の回数に基づき、メニューの切り換えを行う場合のフローチャートである。

【図 1 4】

入力デバイスの特定の操作の継続時間に基づき、メニューの切り換えを行う場合のフローチャートである。

【図 1 5】

メニューをグルーピングして定義する際の設定画面例である。

【図 1 6】

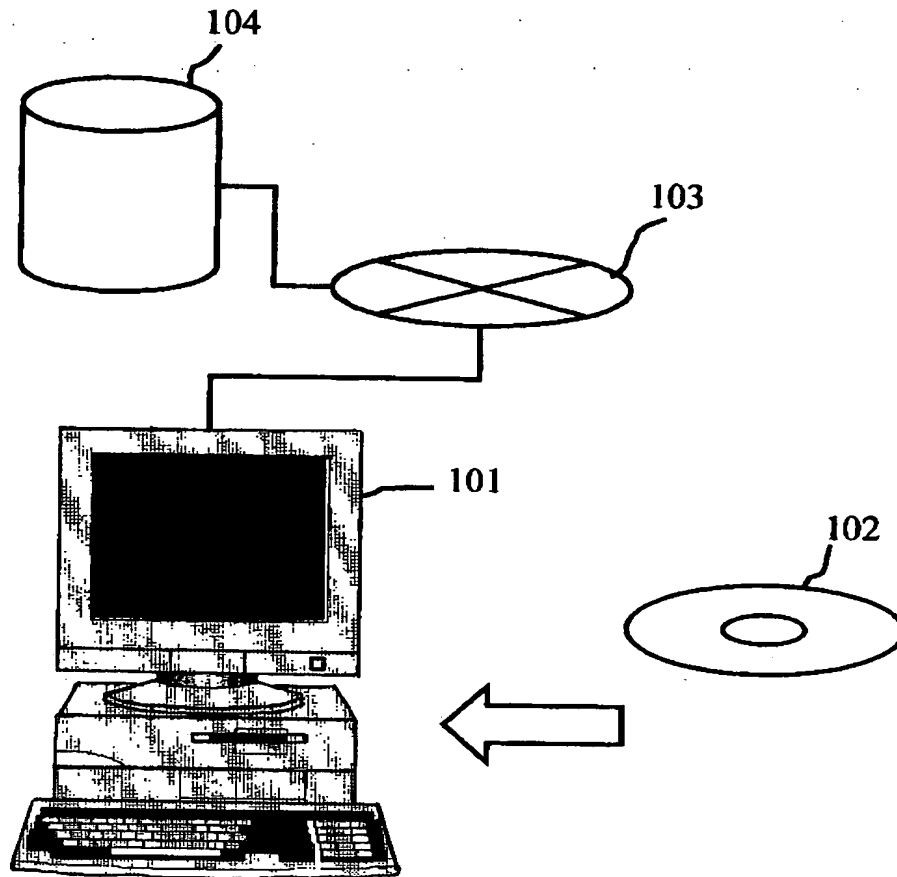
グルーピングして定義された管理手段の一例である。

【図 1 7】

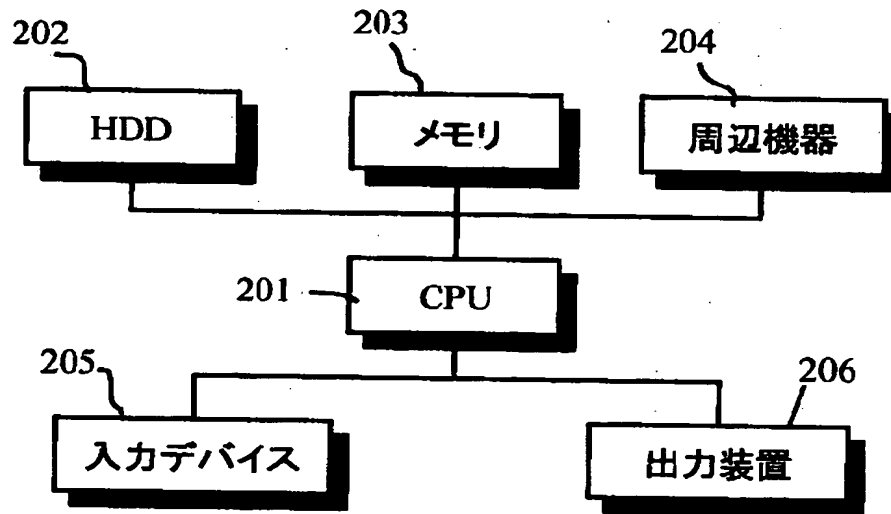
グルーピングして定義された管理手段の他の例である。

【書類名】 図面

【図 1】

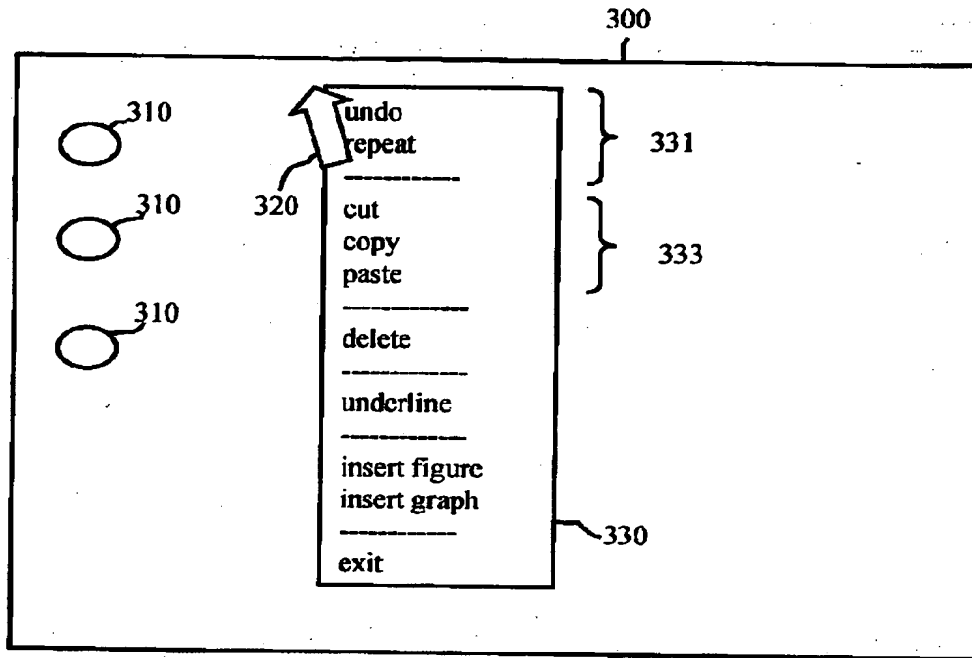


【図 2】



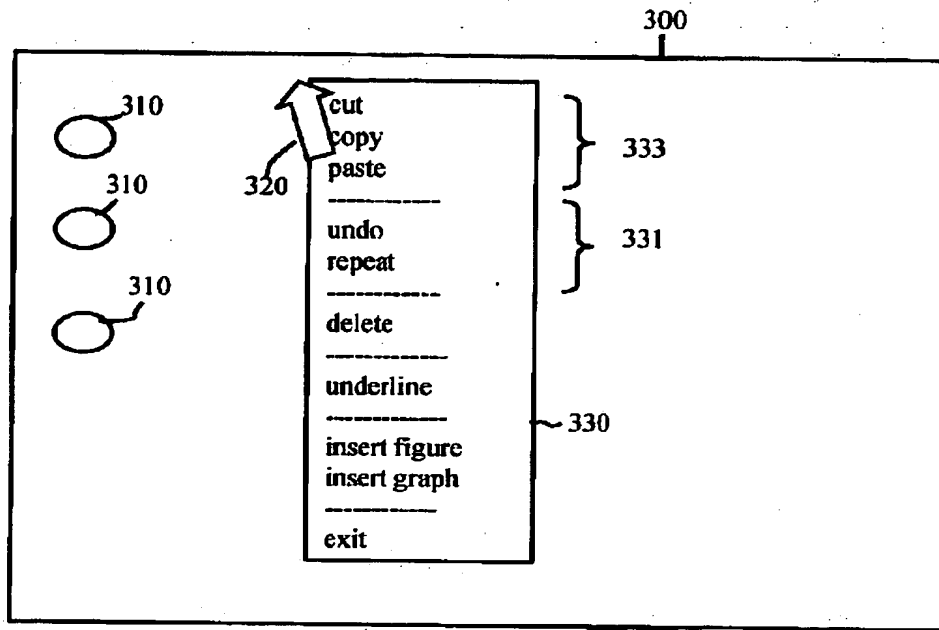
【図 3】

第1の先行技術のGUI



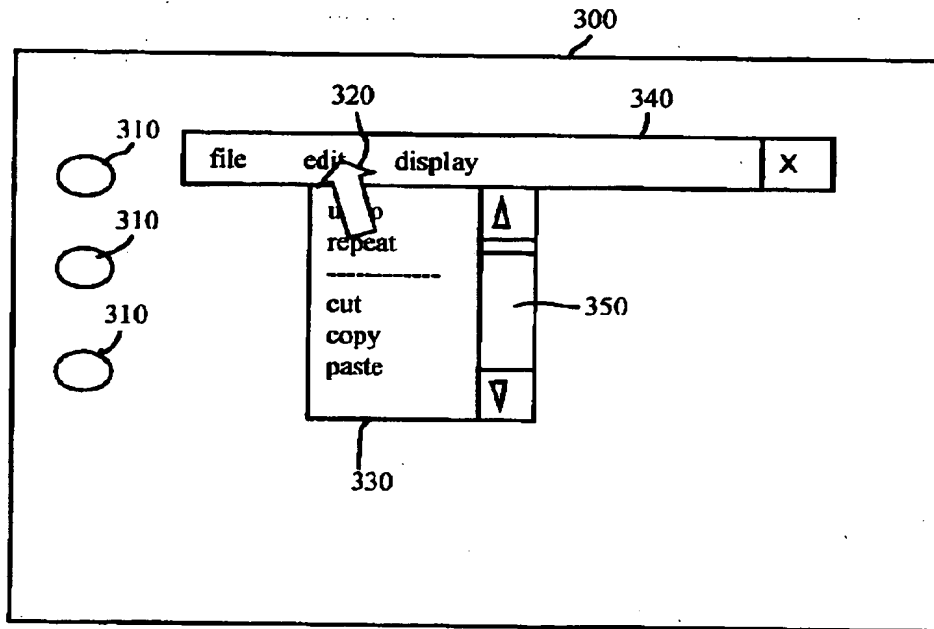
【図 4】

第2の先行技術のGUI



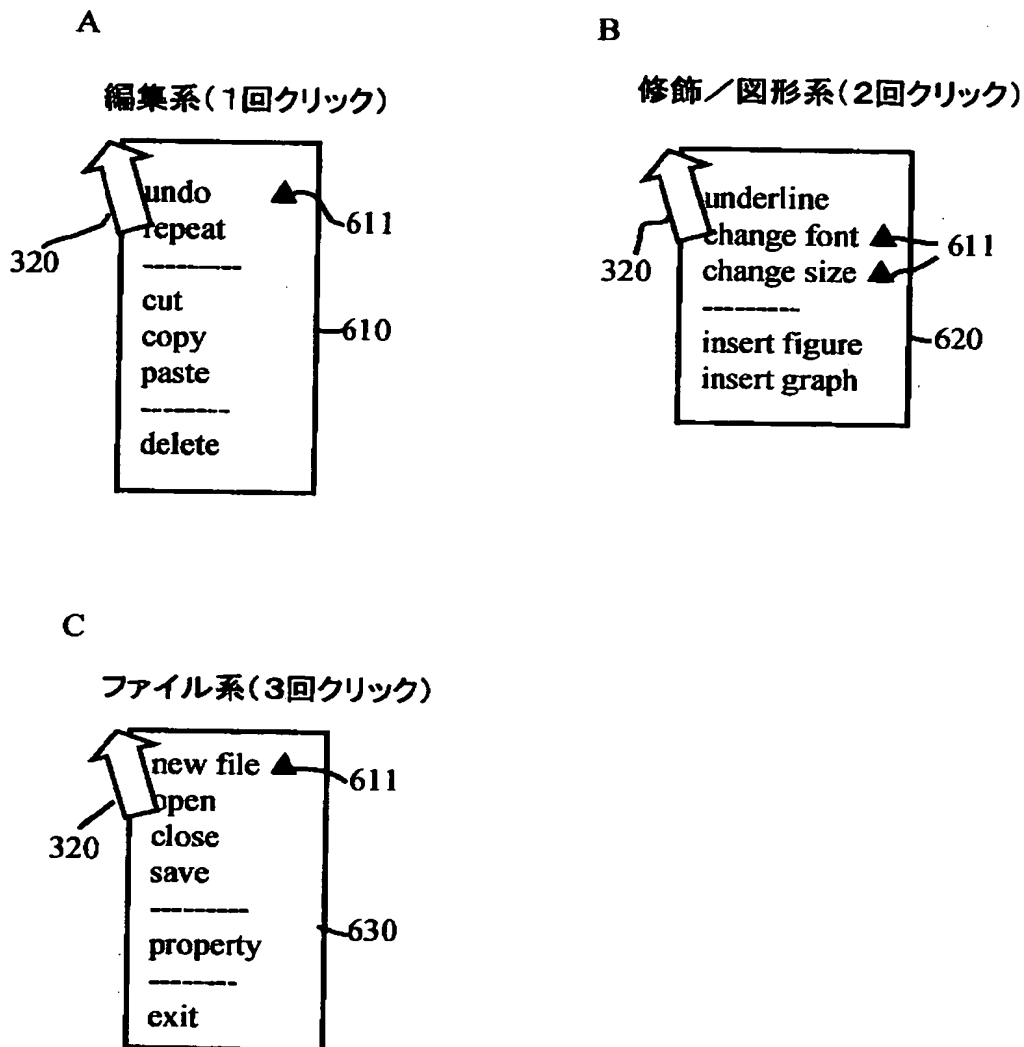
【図 5】

第3の先行技術のGUI



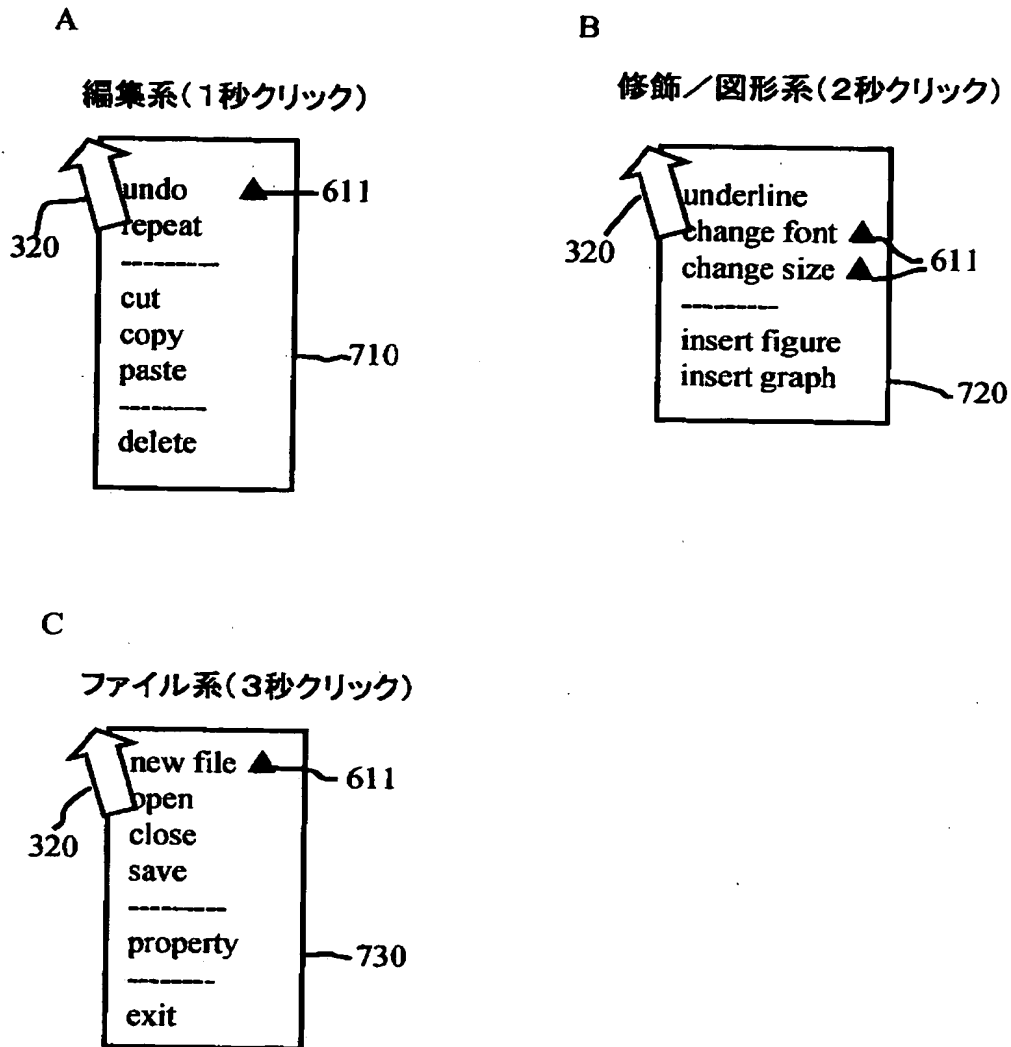
【図 6】

本発明の第1のGUI



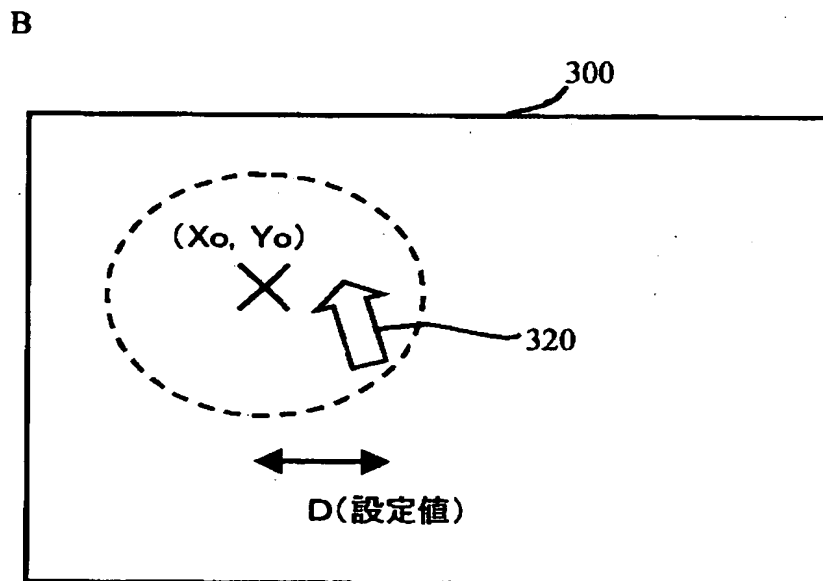
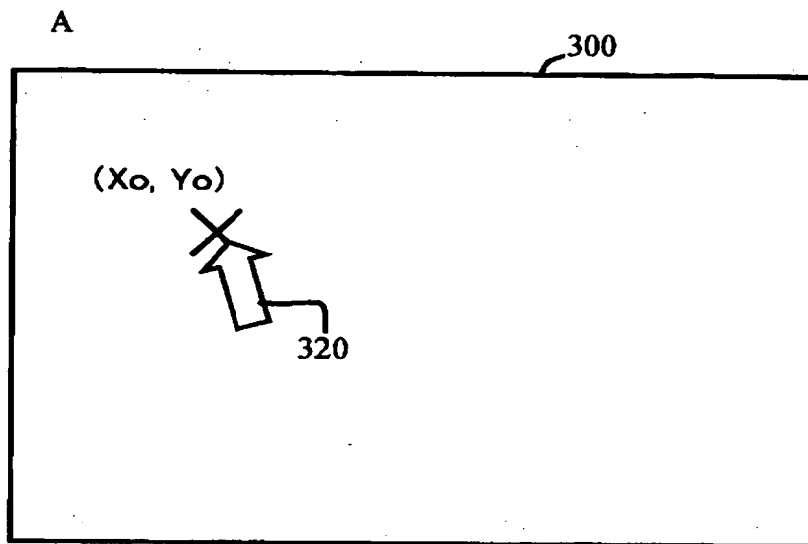
【図 7】

本発明の第2のGUI

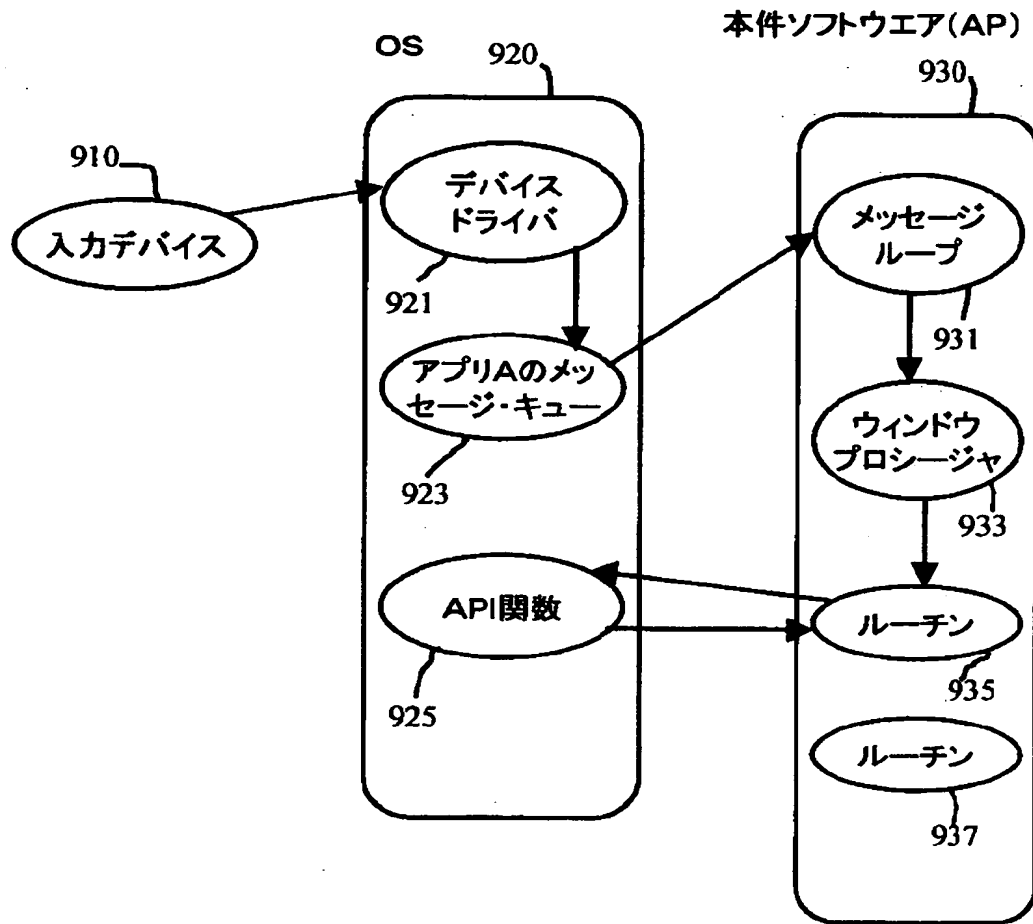




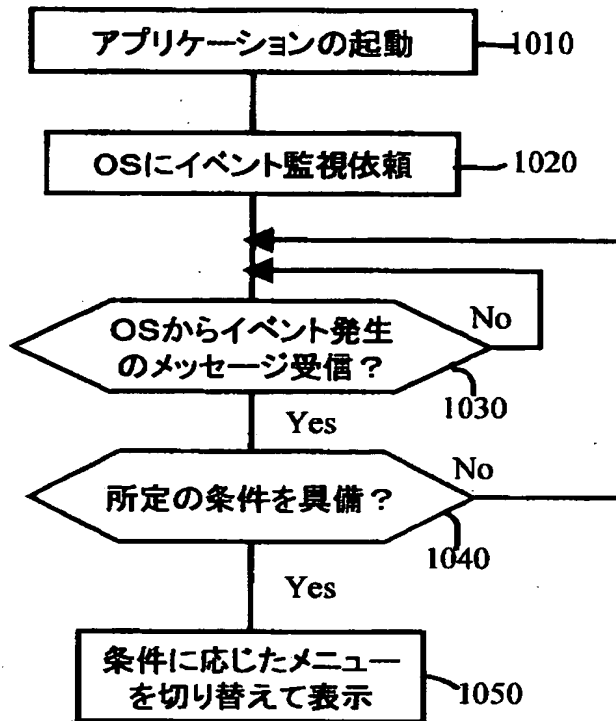
【図 8】



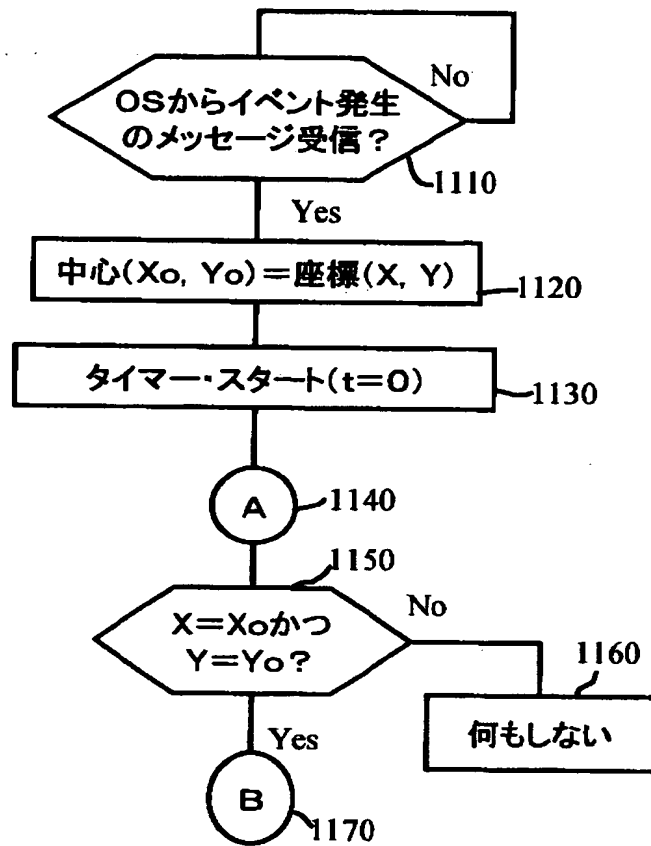
【図 9】



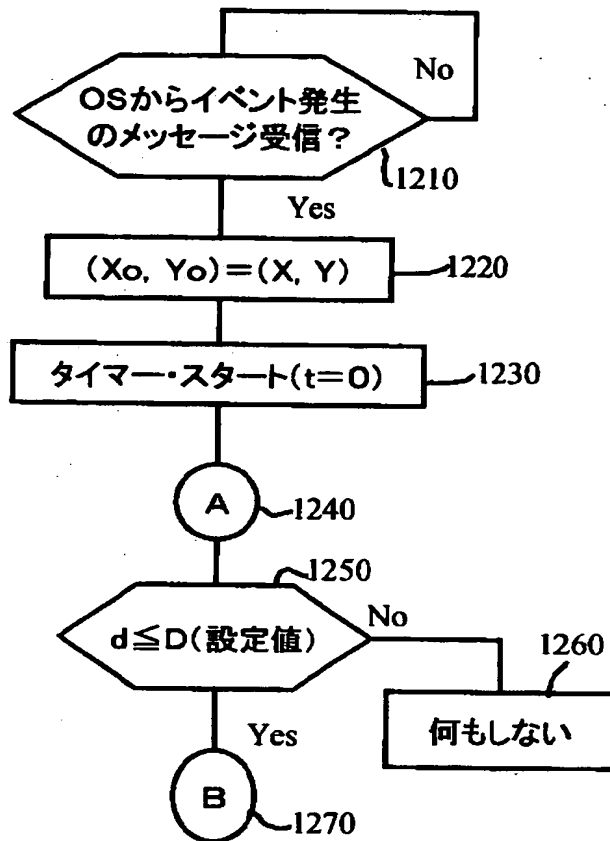
【図 1 0】



【図 1 1】

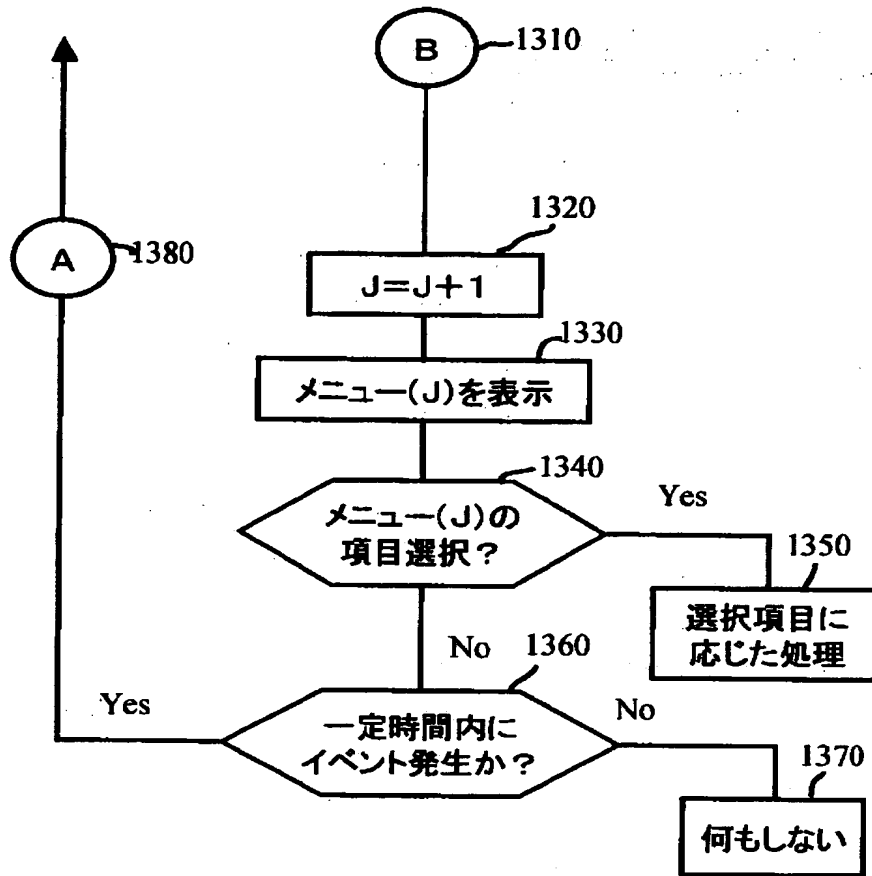


【図 1 2】

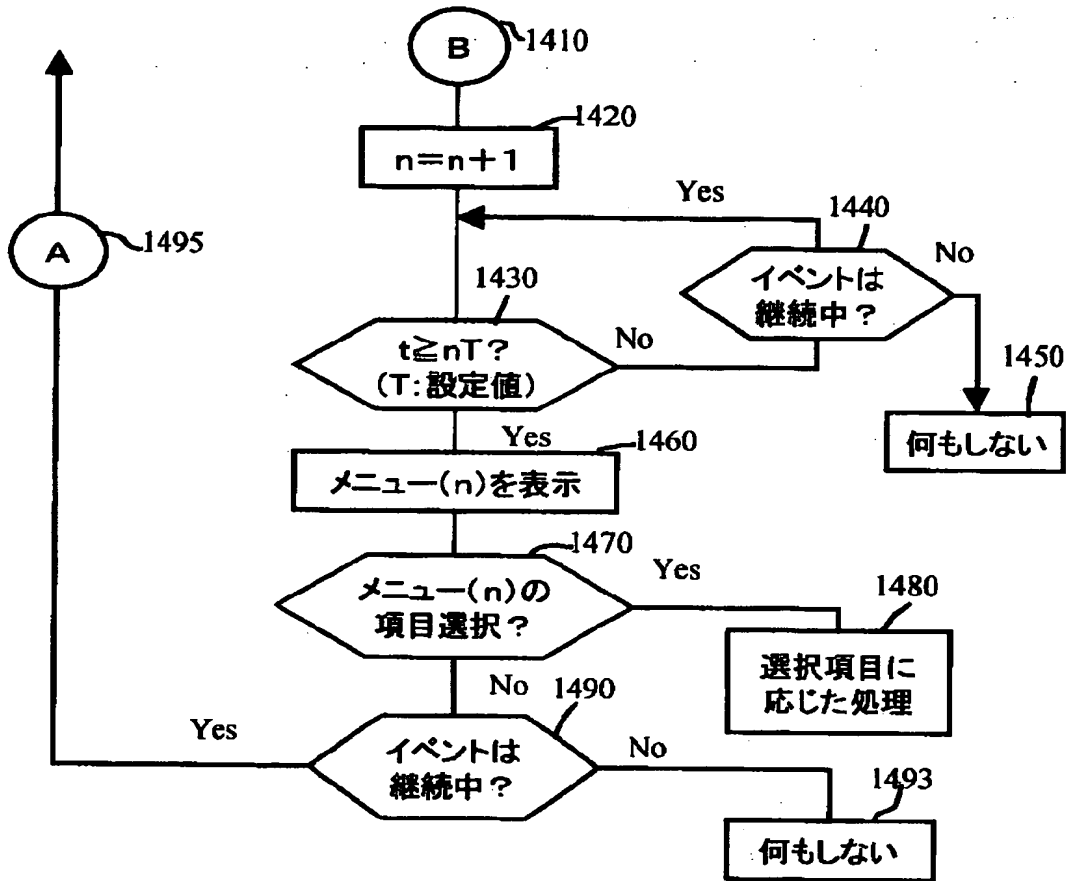


$$d = [(X - X_o)^2 + (Y - Y_o)^2]^{(1/2)}$$

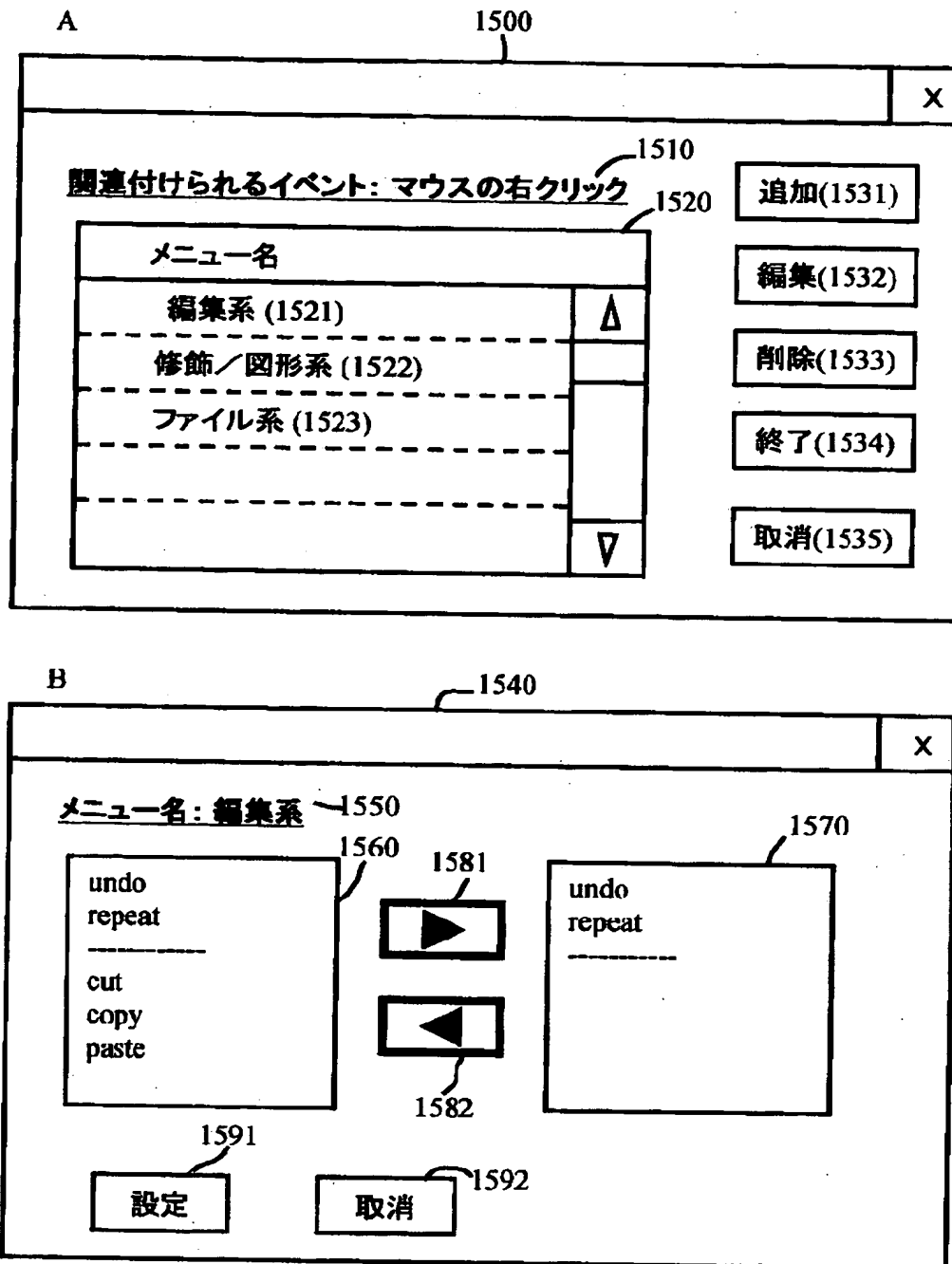
【図 1 3】



【図 1 4】



【図 1 5】





【図 1 6】

A 編集系

1600

```
#include "popup.h" (1601)

BEGIN (1610)
    MENUITEM "&undo", IDM_UNDO (1620)
        BEGIN
            ..... } 1621
        END
    MENUITEM "&repeat", IDM_REPEAT (1622)
    MENUITEM SEPARATOR (1623)
    MENUITEM "&cut", IDM_CUT (1624)
    MENUITEM "&copy", IDM_COPY (1625)
    MENUITEM "&paste", IDM_PASTE (1626)
    MENUITEM SEPARATOR (1627)
    MENUITEM "&delete", IDM_DELETE (1628)
END (1611)
```

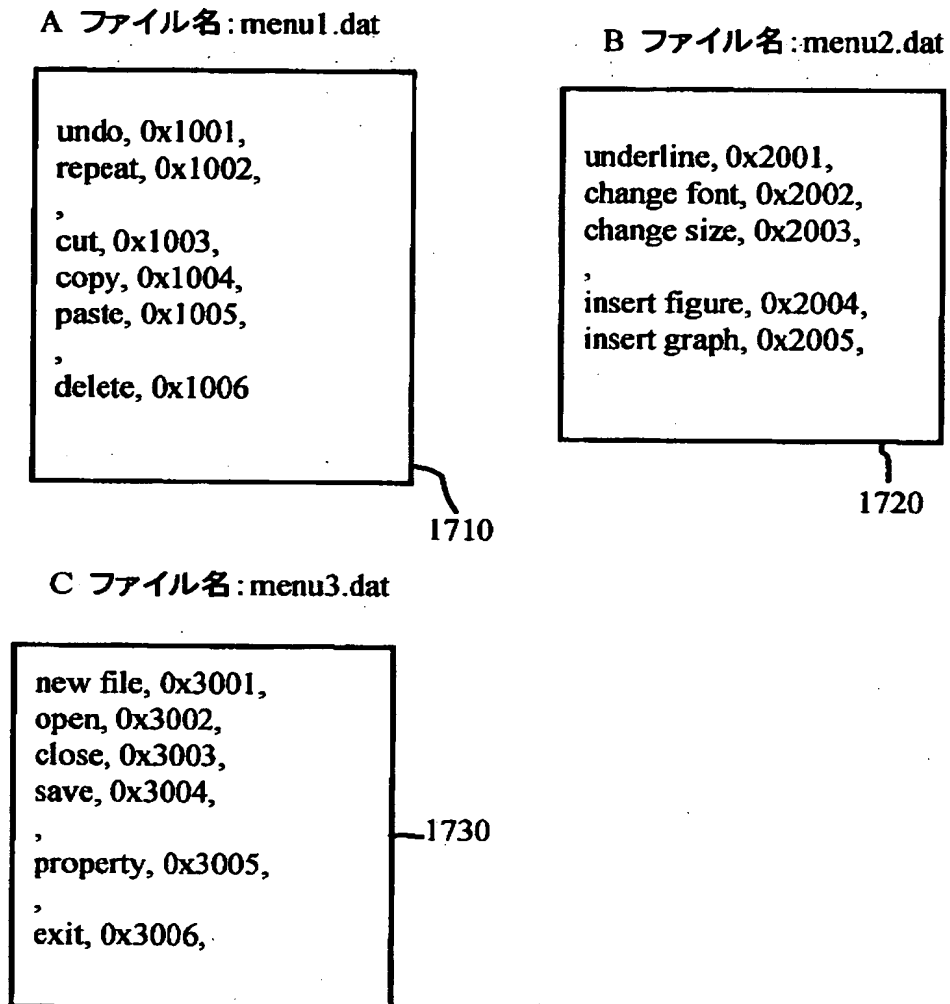
B 修飾／図形系

1650

```
#include "popup.h" (1601)

BEGIN (1610)
    MENUITEM "&underline", IDM_UNDERLINE (1651)
    MENUITEM "&change font", IDM_CHANGEFONT (1652)
        BEGIN
            .... } 1621
        END
    MENUITEM "&change size", IDM_CHANGESIZE (1653)
        BEGIN
            .... } 1621
        END
    MENUITEM SEPARATOR (1654)
    MENUITEM "&insert figure", IDM_INSERTFIGURE (1655)
    MENUITEM "&insert graph", IDM_INSERTGRAPH (1656)
END (1611)
```

【図 1 7】



【書類名】 要約書

【要約】

【目的】 コンピュータシステムにおけるメニューの表示に関し、入力デバイスの操作量（マウスの物理的な移動量のみならず、マウス・ホイールの操作量、トラックパネルの操作量、トラックポイントの操作量、スクロールキーの打鍵回数も含む）を少なくして、ユーザの思考の中断を軽減することを目的とする。

【構成】 本発明を、入力デバイスの操作に基づき表示画面上にメニューを表示するコンピュータシステムにおいて、メニューに表示する項目をグルーピングして管理する管理手段と、入力デバイスの特定の操作回数または特定の操作の継続時間に基づき、前記グルーピングしたメニュー項目を切り替えて表示させる表示制御手段とから構成する。

【選択図】 図 6

出 願 人 履 歴 情 報

識別番号

[000005223]

1. 変更年月日 1996年 3月26日

[変更理由] 住所変更

住 所 神奈川県川崎市中原区上小田中4丁目1番1号

氏 名 富士通株式会社